

# CellML and CESE

by Sergey Missan

## Table of contents

1 Usage.....	2
2 Status.....	2
3 Compatibility.....	3
4 Future work.....	3

The current version of CESE contains a preliminary support for models described using the CellML (<http://www.cellml.org>) format. CellML describes model equations and variables using extensive markup language (derived from the content MathML specification). It also provides support for describing model *structure* (i.e. how model components, such as currents and ionic storages, in case of electrophysiological models) relate to one another, and how they are interacting.

CESE provides means to extract model equations and variables and to convert CellML code into the Java source files, conforming to the JavaBeans specification.

This source code can afterwards be verified, compiled, and packaged similar to the regular CESE model.

## 1. Usage

CellML to CESE model transformation remains extremely experimental at this stage. It is targeted to CESE developers who want to try the conversion routine and improve it. Only partial support for CellML 1.0 is available.

If you want to try the code, download some example CellML models from the CellML repository at <http://www.cellml.org>.

You need to download the XML source code for the model.

Copy the downloaded XML files to `$CESE_HOME/bin/cellml`.

Run a transformation script located in the `$CESE_HOME/` directory, using model's name as a parameter (do not specify the full path), for example:

```
sm:~/sources/cese> ./cellml.sh beeler_reuter_model_1977.xml
Converting: beeler_reuter_model_1977.xml
Source: /home/smissan/sources/cese/bin/cellml/beeler_reuter_model_1977.xml
Destination:
/home/smissan/sources/cese/bin/cellml/beeler_reuter_model_1977.java
```

The model Java source code is now located in the `$CESE_HOME/bin/cellml` directory. Verify the code, and send reports about problems and inconsistencies to [support@simulogic.com](mailto:support@simulogic.com).

## 2. Status

CellML to CESE model transformation uses heavily-modified and updated XSL stylesheet

developed by Mathml-X project. It scans the CellML source code and extracts all variables, wrapping them in getters and setters according to the JavaBeans specification.

The equations are converted from MathML to Java notation, and placed in the correspondent methods. The support for the content MathML is extensive, but still incomplete.

Initial variable values are extracted and properly assigned.

Some basic metadata is extracted and placed in the source code as Java comments.

CellML denotes the hierarchical structure of the model components. Currently, `cellml2Java` stylesheet extracts components as a flat list.

CellML units are not supported.

CellML reactions are not supported.

### **3. Compatibility**

Certain key parameters in CellML models have different names compared to CESE models. These include `V`, `time`, `dtime` (all in CellML notation).

### **4. Future work**

Beta version of the CellML transformer is included in the current version of CESE. BeanInfo creation and packaging is still manual at this stage.